

# GNSS NMEA 0183 标准数据格式的解释和模拟

廖永生 梁绕

(广西第一测绘院 广西南宁 530023)

**【摘 要】** 对 NMEA 0183 格式的定位数据进行解释和数据模拟, 为开发 GNSS 应用服务系统作了基础性研究, 可作为未来 GNSS 各应用系统开发参考。

**【关键词】** NMEA 0183 GNSS 数据 解释 模拟

## 1 前言

GNSS 导航定位技术是目前应用得最广泛的空间定位技术之一, 已被广泛应用于空间信息数据采集和服务等各个方面。随着各区域性 CORS (Continuously Operation Reference System) 建成, 差分 GPS 定位服务得到了不断深化。随着 GLONASS 的完善和 Galileo 卫星导航定位系统的建成, GNSS 导航定位技术将更加普及, 将会对各行各业都产生重大影响。

数据格式问题一直是 GNSS 相关服务中的难题, 特别是差分 GNSS 数据和静态 GNSS 数据格式之间的差异使普通 GPS 设备无法直接获得专业差分服务。数据格式标准的统一, 是实现 GNSS 相关服务的基础。

目前最通用的 GNSS 格式是 NMEA 0183 格式, NMEA 0183 是最终定位格式, 即将二进制定位格式转为统一标准定位格式, 与卫星类型无关。掌握 NMEA 0183 格式, 对于推广 GNSS 应用服务和研究 GNSS 相关技术具有重要意义。

本文将对 NMEA 0183 格式进行概括说明, 同时采用程序模拟 NMEA 0183 格式, 作为 NMEA 0183 标准格式的技术探索。

## 2 NMEA 0183 协议概述

NMEA 是“National Marine Electronics Association”(国际海洋电子协会)的缩写, 同时也是数据传输标准工业协会, 该协会定制的 GNSS 数据格式是 NMEA 0183 数据格式, 它是一套定义接收机输出的标准信息, 有几种不同

的格式，每种都是独立相关的 ASCII 格式，逗号隔开数据流，数据流长度从 30-100 字符不等，通常以每秒间隔选择输出。最常用的格式为“GGA”，它包含了定位时间、纬度、经度、高度、定位所用的卫星数、DOP 值、差分状态和校正时段等，还有速度，跟踪，日期等。

GPS 接收机根据 NMEA 0183 协议的标准规范，将位置、速度等信息通过串口传送到 PC 机、PDA 等设备。NMEA 实际上已成为所有的 GNSS 接收机和最通用的数据输出格式，同时它也被用于与 GNSS 接收机接口的大多数软件包里。

NMEA-0183 协议是 GNSS 接收机应当遵守的标准协议，也是目前 GPS 接收机上使用最广泛的协议，大多数常见的 GNSS 接收机、GNSS 数据处理软件、导航软件都遵守或者至少兼容这个协议。

### 3 NMEA 0183 用途和结构

NMEA 0183 格式与专业 RTCM2.3/3.0 和 CMR+ 的 GNSS 数据格式不同，NMEA 0183 格式主要针对民用定位导航设备，通过 NMEA 0183 格式，可以实现 GNSS 接收机与 PC 或 PDA 之间的数据交换，可以通过 USB 和 COM 口等通用数据接口进行数据传输，其兼容性高，数据传输稳定。

同时 NMEA 0183 可以作为民用差分 GPS 服务解算基础数据，通过 CORS 参考站和 GPRS/CDMA 公网通讯，直接接收 NMEA 0183 兼容格式的差分信息，实现一般差分 GNSS 服务。NMEA 最基本的使用方式如图 1 所示。

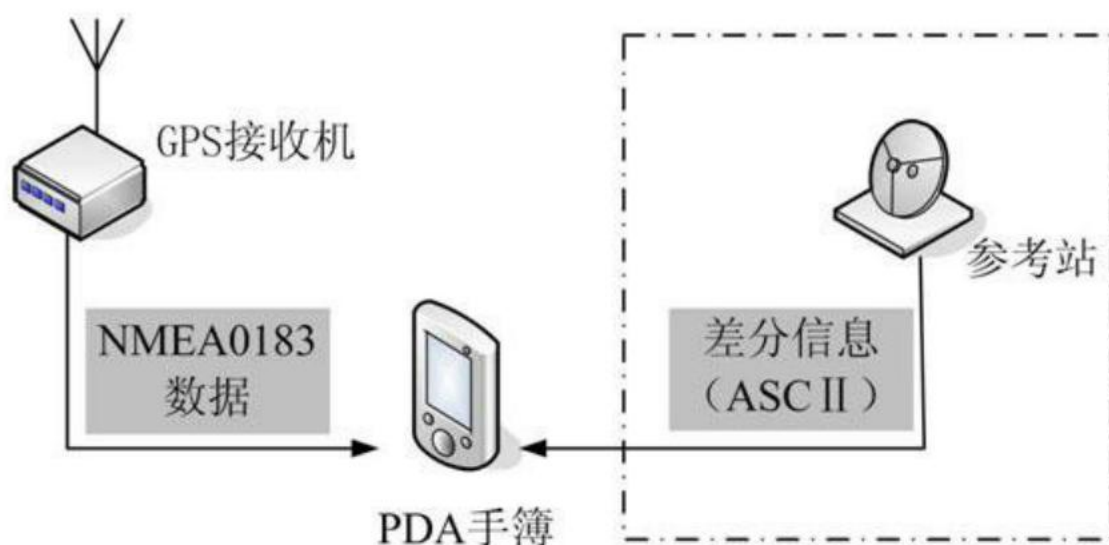


图 1 NMEA 0183 格式基本使用途径

目前以 NMEA 0183 为基础的服务正在应用于车辆导航、船只入港等方面的应用研究。该研究将极大普及差分 GPS 的应用推广，同时扩大 CORS 的应用范围。

4 NMEA 协议的语句

NMEA-0183 协议定义的语句非常多，但是常用的或者说兼容性最广的语句只有\$GPGGA、\$GPGSA、\$GPGSV、\$GPRMC、\$GPVTG、\$GPGLL 等。

NMEA-0183 的每条语句的格式如下表 1 所示。

表 1 NMEA 协议语句格式定义

符号（ASCII）	定义
“\$”	语句起始位
aaccc	地址域，前两位为识别符，后三位为语句名
“， ”	域分隔符
ddd...ddd	发送的数据内容
“*”	效验和符号，后面的两位数是效验和
hh	效验和
<CR>/<LF>	终止符，回车或换行

不同的 GPSOEM 接收板提供的 NMEA 语句有很大差异，主要表现在字段的意义和位置上。用户需要先弄清楚需要哪些信息，然后对照 GPS OEM 接收板的技术资料，寻找那些包括所需信息的 NMEA 语句。

下面将介绍兼容性最广泛的 6 条 NMEA 语句

（1）\$GPGGA 语句

Global Positioning System Fix Data（GGA），即 GPS 定位信息。该语句中反映 GPS 定位主要数据，包括经纬度、质量因子、HDOP、高程、参考站号等字段。

其标准格式如下：

\$GPGGA, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>, <9>, M, <10>, M, <11>, <12>\*hh<CR><LF>

各字段的含义和取值范围见表 2 所示。

表 2 \$GPGGA 语句各字段的含义和取值范围

字段	含义	取值范围
<1>	UTC 时间 hhmmss.ss	000000.00~235959.99
<2>	纬度，格式：ddmm.mmmm	000. 00000~8959.9999
<3>	南北半球	N 北纬；S 南纬
<4>	经度格式 dddmm.mmmm	00000.0000~17959.9999
<5>	东西半球	E 表示东经；W 表示西经
<6>	GPS 状态	0=未定位，1=GPS 单点定位固定解，2=差分定位，3=PPS 解；4=RTK 固定解；5=RTK 浮点解；6=估计值；7=手工输入模式；8=模拟模式
<7>	应用解算位置的卫星数	00~12
<8>	HDOP， 水平图形强度因子	0.500~99.000（大于 6 不可用）
<9>	海拔高度	-9999.9~99999.9
<10>	地球椭球面相对大地水准面的高度（高程异常）	-9999.9~99999.9
<11>	差分时间	从最近一次接收到差分信号开始的秒数，如果不是差分定位将为空
<12>	参考站号	0000~1023；不使用 DGPS 时空

例句：

\$GPGGA, 074529.82, 2429.6717, N, 11804.6973, E, 1, 8, 1.098, 42.110, M, , M, , \*76

（2）\$GPGSA 语句

GPS DOP and Active Satellites（GSA）即当前卫星信息， 其标准格式如下：

\$GPGSA, <1>, <2>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <3>, <4>, <5>, <6>\*hh<CR><LF>

各字段的含义和取值范围见表 3 所示。

表 3 \$GPGSA 语句各字段的含义和取值范围



字段	含义	取值范围
<1>	模式	M=手动，A=自动
<2>	定位类型	1=没有定位，2=2D 定位，3=3D 定位
<3>	PRN 码（伪随机噪声码），正在用于解算位置的卫星号	01~32
<4>	PDOP 位置精度因子	0.5~99.9
<5>	HDOP 水平精度因子	0.5~99.9
<6>	VDOP 垂直精度因子	0.5~99.9

例句：

\$GPGSA, A, 3, 19, 11, 03, 23, 27, 13, 16, , , , , , 3.43, 1.67, 2.99\*0E

### （3）\$GPGSV 语句

GPS Satellites in View（GSV）可见卫星信息，反映 GPS 可见星的方位角、俯仰角、信噪比等。

其标准格式如下：

\$GPGSV, <1>, <2>, <3>, <4>, <5>, <6>, <7>, ...<4>, <5>, <6>, <7>\*hh<CR><LF>

各字段的含义和取值范围见表 4 所示。

表 4 \$GPGSV 语句各字段的含义和取值范围

字段	含义	取值范围
<1>	总的 GSV 语句电文数	0~12
<2>	当前 GSV 语句号	1-3
<3>	可视卫星总数	00~12
<4>	卫星号	01~32
<5>	卫星仰角	00~90 度
<6>	卫星方位角	000~359 度
<7>	信噪比	00~99dB，没有跟踪到卫星时为空

注：<4>，<5>，<6>，<7>信息将按照每颗卫星进行循环显示，每条 GSV 语句最多可以显示 4 颗卫星的信息。

其他卫星信息将在下一序列的 NMEA0183 语句中输出。

例句：

\$GPGSV, 3, 1, 11, 1, 83, 54, 32, 3, 19, 192, 28, 6, 26, 57, 36, 7, 51, 140, 37\*7D

\$GPGSV, 3, 2, 11, 14, 40, 136, 34, 16, 64, 266, 36, 20, 21, 293, , 22, 2, 168, \*4C

\$GPGSV, 3, 3, 11, 23, 10, 321, , 25, 53, 7, 40, 30, 2, 46, \*48

(4) \$GPRMC 语句

Recommended Minimum Specific GPS/TRANSIT Data (RMC)，即推荐定位信息。

其标准格式如下：

\$GPRMC, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>, <9>, <10>, <11>, <12>\*hh<CR><LF>

各字段的含义和取值范围见表 5 所示。

表 5 \$GPRMC 语句各字段的含义和取值范围

字段	含义	取值范围
<1>	UTC 时间， hhmmss.ss	000000.00~235959.99
<2>	定位状态	A=有效定位，V=无效定位
<3>	纬度，格式：ddmm.mmmm	000. 00000~8959.9999
<4>	南北半球	N 表示北纬；S 表示南纬
<5>	经度格式 dddmm.mmmm	00000.0000~17959.9999
<6>	东西半球	E 表示东经；W 表示西经
<7>	地面速率	000.0~999.9 节
<8>	地面航向	000.0~359.9，以真北为参考基准
<9>	UTC 日期，格式：ddmmyy	日月年
<10>	磁偏角	000.0~180.0 度
<11>	磁偏角	E 表示偏东；W 表示偏西
<12>	模式指示(NMEA0183 3.00 版本输出)	A=自主定位，D=差分，E=估算，N=数据无效

例句：

\$GPRMC, 074529.82, A, 2429.6717, N, 11804.6973, E, 12.623, 32.122, 010806, , W, A\*08

(5) \$GPVTG 语句

Track Made Good and Ground Speed（VTG），即地面速度信息。其标准格式为：

\$GPVTG, <1>, T, <2>, M, <3>, N, <4>, K, <5>\*hh<CR><LF>

各字段的含义和取值范围见表 6 所示。

表 6 \$GPVTG 语句各字段的含义和取值范围

字段	含义	取值范围
<1>	以真北为参考基准的地面航向	000~359 度
<2>	以磁北为参考基准的地面航向	000~359 度
<3>	地面速率	000.0~999.9 节
<4>	地面速率	0000.0~1851.8 公里/小时
<5>	模式指示（仅 NMEA0183 3.00 版本输出）	A=自主定位，D=差分，E=估算，N=数据无效

例句：

\$GPVTG, 257.314, T, 257.314, M, 10.739, N, 19.888, K, A\*2F

(6) \$GPGLL 语句

Geographic Position（GLL），即定位地理信息。其标准格式如下：

\$GPGLL, <1>, <2>, <3>, <4>, <5>, <6>, <7>\*hh<CR><LF>

各字段的含义和取值范围见表 7 所示。

表 7 \$GPGLL 语句各字段的含义和取值范围

字段	含义	取值范围
<1>	纬度，格式：ddmm.mmmmm	000. 00000~8959.9999
<2>	南北半球	N 表示北半球；S 表示南半球

<3>	经度，格式：dddmm.mmmm	00000.0000~17959.9999
<4>	东西半球	E 表示东经；W 表示西经
<5>	UTC 时间，格式：hhmmss	000000.00~235959.99
<6>	定位状态	A=有效定位，V=无效定位
<7>	模式指示(NMEA0183 3.00 版本输出)	A=自主定位，D=差分，E=估算，N=数据无效

## 5 NMEA 0183 数据模拟

该源代码为 VC++源代码，是对一个模拟 NMEA 数据的类的定义。

```
struct CNmeaData
```

```
{
```

```
CNmeaData () ;
```

```
void ResetData () ;
```

```
// Data retrieved from the NMEA sentences.
```

```
double lat;          // 纬度，单位为度（正数为北纬，负数为南纬） double lon;          // 经度，单位为度（正
```

数为东经，负数为西经）

```
double altitude;     // 海拔高度，米
```

```
double speed;        // 速度，节
```

```
double track;        //追踪卫星高度角，度
```

```
double magVariation; // 磁偏角，度
```

```
double hdop;         // HDOP 值
```

```
int numSats;         // 追踪卫星数
```

```
int UTCYear;         // UTC 年
```

```
int UTCMonth;        // UTC 月
```

```
int UTCDay;          // UTC 日
```



```

int UTCHour;        // UTC 时

int UTCMinute;      // UTC 分

int UTCSecond;      // UTC 秒

CSatData satData[MAC_SATS];

// 卫星锁定数:

// 0 = 无效, 1 = 卫星锁定, 2 = DGPS 状态

GPS_FIX_QUALITY lastFixQuality;

// 有效卫星数

bool isValidLat;     // 纬度

bool isValidLon;     // 经度

bool isValidAltitude; // 高程

bool isValidSpeed;   // 速度

bool isValidDate;    // 日期

bool isValidTime;    // 时间

bool isValidTrack;    // 追踪卫星角

bool isValidMagVariation; // 磁偏角变化

bool isValidHdop;     // HDOP 值

bool isValidSatData;  // 卫星数

// 确定是否有有效的坐标数据发送到数据端口

bool hasCoordEverBeenValid;

};

class CNmeaParser

```

```

{

public:

    CNmeaParser () ;

    SENTENCE_STATUS ParseSentence (const char* sentence) ;

    void GetData (CNmeaData& data) const;

    void ResetData () {m_data.ResetData () ; }

private:

    bool ParseDegrees (double& degrees, const char* degString) const;

    bool ParseDate (int& year, int& month, int& day,

const char* dateString) const;

    bool ParseTime (int& hour, int& minute, int& second,

const char* timeString) const;

    void ParseAndValidateAltitude (const char* field, const char unit) ;

    void ParseAndValidateDate (const char* field) ;

    void ParseAndValidateFixQuality (const char* field) ;

    void ParseAndValidateLat (const char* field, const char hem) ;

    void ParseAndValidateLon (const char* field, const char hem) ;

    void ParseAndValidateHdop (const char* field) ;

    void ParseAndValidateSpeed (const char* field) ;

    void ParseAndValidateMagVariation (const char* field,

const char direction) ;

    void ParseAndValidateTime (const char* field) ;

```

```

void ParseAndValidateTrack (const char* field) ;

void ParseGGA (const char* sentence) ;

void ParseGLL (const char* sentence) ;

void ParseRMC (const char* sentence) ;

void ParseGSV (const char* sentence) ;

bool GetNextField (char* data, const char* sentence,
uint& currentPosition) const;

bool IsValidSentenceType (const char* sentence) const;

bool IsCorrectChecksum (const char* sentence) const;

CNmeaData m_data;

// 需要分解 GSV 语句

int m_lastSentenceNumber; // 确定结束语句

int m_numSentences;      // 处理语句数

int m_numSatsExpected;    // 分析捕获卫星数

int m_numSatsLeft;        // 分析失锁卫星数

int m_satArrayPos;        // 下一次捕获卫星位置预测

CSatData m_tempSatData[MAC_SATS];

};

class CNmeaSerial

{public:

CNmeaSerial () ;

CNmeaSerial (HWND hWnd, DWORD timeout) ;

```

```

~CNmeaSerial ( ) ;

SERIAL_RESULT_TYPE CloseConnection ( ) ;

SERIAL_RESULT_TYPE OpenConnection ( ) ;

SERIAL_RESULT_TYPE OpenConnection (uint8 comPort, uint32 baudRate,
uint8 dataBits, SETPARITY parity, STOPBITS stopBits) ;

SERIAL_RESULT_TYPE GetData (CNmeaData& data) const;

SERIAL_RESULT_TYPE GetSettings (uint8& comPort, uint32& baudRate,
uint8& dataBits, SETPARITY& parity, STOPBITS& stopBits) const;

unsigned long GetThreadHandle ( ) const {return m_hThread; }

BOOL IsConnected ( ) const {return m_isConnected; }

private:

SERIAL_RESULT_TYPE InitThread ( ) ;

BOOL KillThread ( ) ;

BOOL LockData ( ) const;

void SetDefaults ( ) ;

BOOL UnlockData ( ) const;

// 元函数描述语句，并运行该语句

// 进入另外线程

static void ProcessSentences (void* currentObject) ;

uint8 m_comPort;

uint32 m_baudRate;

uint8 m_dataBits;

```

```

SETPARITY m_parity;

STOPBITS m_stopBits;

BOOL m_isConnected;

DWORD m_timeout;

unsigned long m_hThread;

HANDLE m_hPort;

HANDLE m_hMutex;

DWORD m_threadId;

CNmeaParser m_parser;

// 接收描述信息窗口

HWND m_hMsgWnd;

// 如果值为 True，则进行线程分析

// 否则结束状态

BOOL m_terminateThread; }

```

掌握 NMEA 0183 数据的基本原理和格式，并实现模拟 NMEA 0183 数据，对于开发 GNSS 服务和应用系统，扩展 GNSS 服务范围和服务方式等，都有重要意义。